(12) UK Patent Application (19) GB (11) 2 357 348 (13) A

(43) Date of A Publication 20.06.2001

(21) Application No 9929936.4

(22) Date of Filing 18.12.1999

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
John Bryan Ibbotson
Stephen James Paul Todd

(74) Agent and/or Address for Service
K J Fournier
IBM United Kingdom Limited, Intellectual Property
Department, Hursley Park, WINCHESTER, Hampshire,
SO21 2JN, United Kingdom

(51) INT CL^7
G06F 17/27 17/30

(52) UK CL (Edition S)
G4A AKBX AUDB

(56) Documents Cited
WO 98/53393 A1
"Class com.studiom.dom.DOM", 26 Aug '99, at
www.studiom.com/sw/dom/docs/api/com.
studiom.dom.DOM.html

(58) Field of Search
UK CL (Edition R) G4A AKBX AUDB
INT CL^7 G06F 17/21 17/22 17/27 17/30
Online: COMPUTER, EPODOC, INSPEC, Internet,
JAPIO, TDB, WPI

(54) Abstract Title
Using an abstract messaging interface and associated parsers to access standard document object models

(57) A data processing apparatus has: a memory unit 402; a system software unit 401 including a plurality of parsers 401A, 401B, for implementing a Document Object Mode (DOM) application programming interface, including a unit for receiving documents from an originating application 11, 21 and a unit for processing received documents using one of the parsers in order to convert each received document into an object model; and a system software unit for storing each object model into the memory unit; where the unit for processing sends a received document to a particular one of the plurality of parsers depending on the format of the received document. This allows the DOM to accept documents in formats other than XML/HTML, e.g in COBOL.

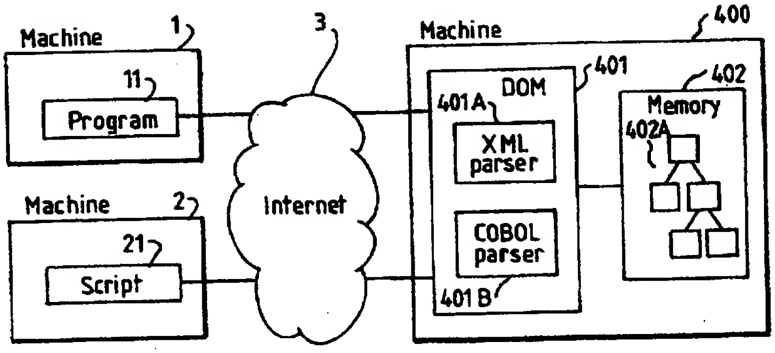

FIG. 2

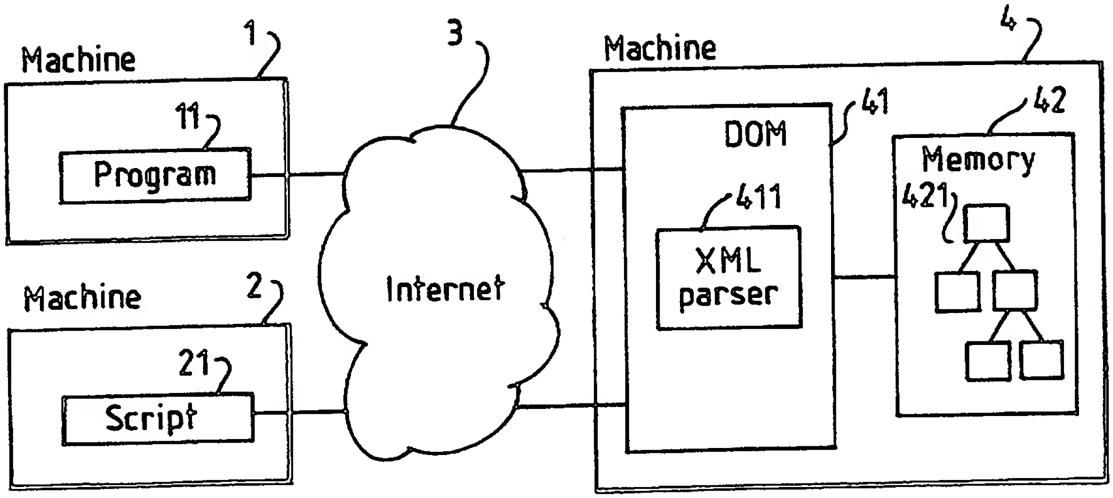

FIG. 1 PRIOR ART

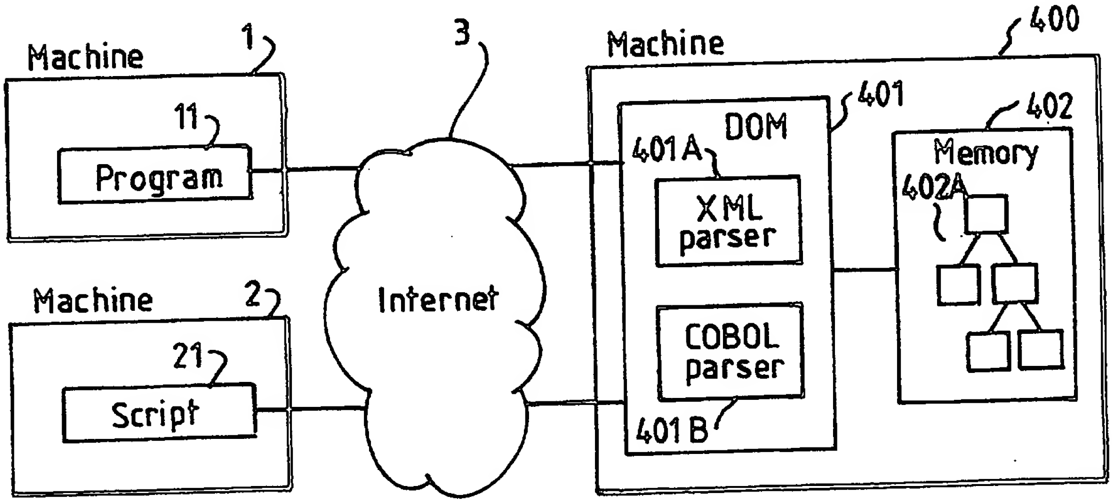

FIG. 2

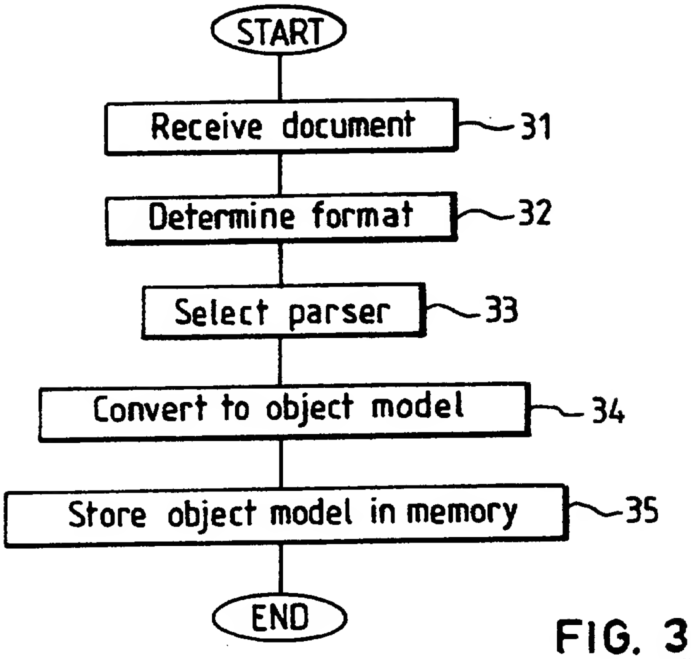
START
Receive document
31
Determine format
32
Select parser
33
Convert to object model
34
Store object model in memory
35
END

FIG. 3

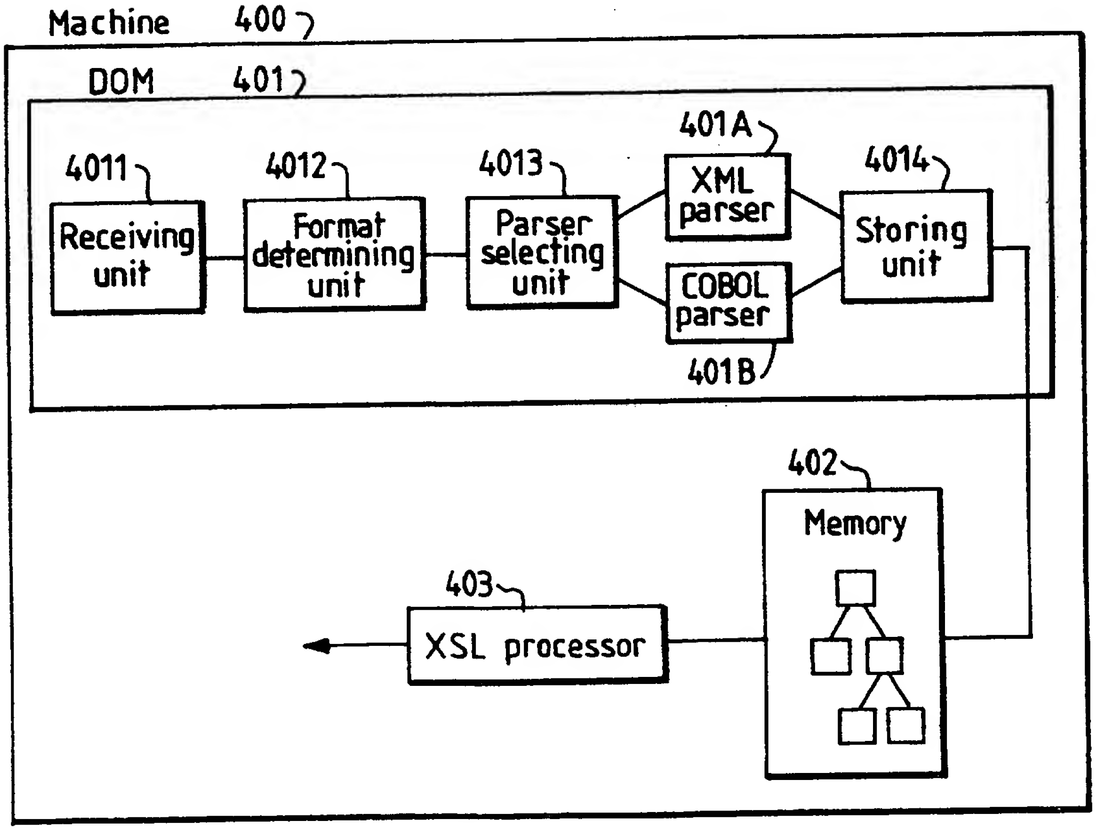
Machine
400
DOM
401
4011
Receiving unit
4012
Format determining unit
4013
Parser selecting unit
401A
XML parser
COBOL parser
401B
4014
Storing unit
402
Memory
403
XSL processor

FIG. 4

USING AN ABSTRACT MESSAGING INTERFACE AND ASSOCIATED PARSERS TO ACCESS STANDARD DOCUMENT OBJECT MODELS

Field of the Invention

The present invention relates to the field of data processing, and more particularly to the art of computer software programming making use of a Document Object Model application programming interface.

Background of the Invention

The Document Object Model (DOM) is a platform-neutral and language-neutral application programming interface (API) that will allow programs and scripts to dynamically access and update the content, structure and style of HTML and XML documents. The World Wide Web Consortium (W3C) has developed published specifications for the DOM and an important objective of the W3C is to provide a standard programming interface that can be used in a wide variety of environments and applications. See, e.g., "What is the Document Object Model?" by J. Robie of Texcel Research, pp. 1-4, printed from the World Wide Web on November 4, 1999 at

http://www.w3.org/TR/REC-DOM-Level-1/introduction.html

such document is herein incorporated by reference.

Fig. 1 is an example of the use of prior art DOM technology. A first data processing machine 1 (e.g., a personal computer running a Web browser) is running a program 11 and a second data processing machine is running a script 21. A third data processing machine 4 is running the DOM software 41. The machines 1 and 2 are communicating with the machine 4 via the Internet 3 using normal World Wide Web technology (i.e., TCP/IP and HTTP). The DOM 41 receives XML documents from the program 11 and script 21 and converts each such XML document into a Java object model, which is a group of objects, which is then stored in memory 42 at the data processing machine 4 where the DOM software 41 is running. This Java object model is, in effect, a tree 421 of nodes containing the data and structure which was contained in the XML document that was originally received by the DOM. The XML documents which arrive at the DOM 41 are parsed by an XML parser 411 in order to convert them into respective Java object models which are then stored in memory 42. Once the Java object models are stored in memory 42, the program 11 and script 21 (or any other program or script that can access the DOM 41) uses the DOM 41's API to access and modify the Java object models of the XML documents that have been previously received and stored in memory 42.

While the DOM allows programs and scripts to dynamically access and update the content, structure and style of documents, the documents which the DOM will accept are quite limited in that they must be in XML (or in HTML). This has greatly limited the use of the DOM and it would be very advantageous for programs or scripts to be able to provide non-XML/HTML documents to a DOM. This would allow the DOM to be much more versatile. However, according to the present state of the art, there has been no solution to this problem provided in the marketplace.

Summary of the Invention

According to a first aspect, the present invention provides a data processing apparatus having: a memory unit; a system software unit, including a plurality of parsers, for implementing a Document Object Model application programming interface, including a unit for receiving documents from an originating application and a unit for processing received documents using one of the plurality of parsers in order to convert each received document into an object model; and a system software unit for storing each object model into the memory unit; wherein the unit for processing sends a received document to a particular one of the plurality of parsers depending on a format of the received document.

According to a second aspect, the present invention provides a data processing method having steps of: receiving a document from an originating application via a Document Object Model application programming interface which includes a plurality of parsers; determining a format of the received document; selecting one of the plurality of parsers depending on the results of the determining step; converting the received document into an object model using the parser selected at the selecting step; and storing the object model into a memory unit.

According to a third aspect, the present invention provides a computer program product stored on a computer readable storage medium for, when run on a computer, instructing the computer to carry out the method steps of the second aspect.

According to a fourth aspect, the present invention provides a data processing apparatus having functional processing components for carrying out the respective steps of the method of the second aspect.

Thus, with the present invention, the usefulness of the DOM is greatly improved since documents of different formats can be accepted. The DOM is no longer limited to dealing with only XML/HTML format documents. Preferably, through the use of an XSL processor placed after the memory unit, non-XML/HTML documents which were originally received by the DOM and stored in the memory unit can be transformed into an output form for display using a Web browser.

Brief Description of the Drawings

The present invention will be better understood upon reading the below described detailed description of the preferred embodiments thereof, which will be presented with reference to the following drawings:

Fig. 1 is a block diagram showing three data processing machines running software according to the prior art;

Fig. 2 is a block diagram showing three data processing machines running software according to a preferred embodiment of the present invention;

Fig. 3 is a flowchart showing the operational steps performed by the system software of machine 400 of Fig. 2, according to a preferred embodiment of the present invention; and

Fig. 4 is a block diagram showing functional software blocks for carrying out the functionality of a preferred embodiment of the present invention.

Detailed Description of the Preferred Embodiments

In Fig. 2, the same basic architecture is illustrated as was shown and described above with respect to Fig. 1. However, the DOM 401 according to a preferred embodiment of the present invention includes more parsers than only the single XML parser 411 that was shown in Fig. 1. For example, in Fig. 2, a COBOL parser 401B is shown in addition to the XML parser 401A. The term "COBOL parser" is used here to mean a parser capable of parsing a structured message such as is typically described in a COBOL Copybook.

If an XML format document is received over Internet 3 from program 11, the system software running in machine 400 which implements the DOM 401 recognizes the format of the document as the XML format and thus routes the received XML format document to the XML parser 401A. On the other hand, if a COBOL format document is received over Internet 3 from program 11, the system software running in machine 400 which implements the DOM 401 recognizes the format of the document as the COBOL format and thus routes the received COBOL format document to the COBOL parser 401B.

Preferably, the system software code used to implement the DOM 401 having a plurality of parsers is the code which is being used in message brokers. Specifically, such a message broker receives messages in a variety of formats and then converts the received messages into an abstract message model (i.e., object model). The format of a received message is determined and then a particular parser of a group of parallel parsers is selected based on the determined format of the message for processing the received message in order to convert the received message into the abstract message model. For example, SAGA Software Inc. of Reston Virginia USA have such a message broker product called Sagavista (TM). See, e.g., the World Wide Web-published white paper "Sagavista (TM) Expanding the Reach of Your Enterprise" by David S. Linthicum, Chief Technology Officer, SAGA, such document being herein incorporated by reference.

In these message brokers, each received message has associated with it some meta-information describing the received message's structure, content and physical representation (referred to herein as the message's "format"). This format information is used to select an appropriate parser (from a plurality of parallel parsers) which converts the message to the common abstract message model. Once in this common abstract message model, each message can be processed using the same processing nodes within the message broker even though the messages have originated from different formats. The format also defines the mapping from the abstract message model to the physical representation (e.g., record datastructure (Cobol or C), XML tagged structure, etc.).

Steps explaining the operation of the system software code running in machine 400 will now be described with reference to the flowchart of Fig. 3 to illustrate the operation of a DOM that can handle documents in a plurality of formats.

At step 31, the system software code receives a document over the Internet 3 from program 11 running in machine 1. At step 32, the format of the document is determined. At step 33, a parser is selected from a group of parsers, with the selection being dependent on the determined format of the document. At step 34, the selected parser converts the document into an object model (i.e., a tree of nodes). Finally, at step 35, the object model is stored in memory 402 (an object model 402A is shown in memory 402 of Fig. 2). In accordance with existing parsing art, steps 34 and 35 may alternatively be performed in a "lazy" manner thus making them "lazy parsers". Specifically, only those parts of the received document that are requested by the DOM interface, or those parts of the document that must be parsed in order to access the requested parts, are parsed and converted to the object model in step 34.

In Fig. 4, the system software, according to a preferred embodiment of the present invention, running on machine 400 of Fig. 2 includes the DOM 401 including the two parsers 401A and 401B as described above. The DOM 401 also includes a receiving unit 4011 which receives documents that were sent by, for example, program 11 on machine 1 over the Internet 3. The received documents are sent to a format determining unit 4012 which determines the format of the received documents. The documents are then forwarded to the parser selecting unit 4013 which selects one of the parsers 401A or 401B depending on whether the format determining unit 4012 has determined that the format of the received document is XML or COBOL. It should be noted that while only two parsers are illustrated in this example, in practice there would probably be more than two used. The parsers convert the document into the object model, as described above, and a storing unit 4014 then stores the object model into the memory 402. An XSL processor 403 then performs a transformation on the stored object models in order to render them in a form in which they can be displayed on a display screen, preferably using a Web browser.

XML (eXtensible Markup Language) emphasizes description of information structure and content as distinct from its presentation. The data structure and its syntax are defined in a DTD (Document Type Definition) specification, which is a derivative from SGML and defines a series of tags and their constraints. In contrast to information structure, the presentation issues are addressed by XSL (eXtensible Style Language), which is also a W3C emerging standard for defining how XML-based data should be expressed on a display screen (e.g., by a Web browser).

XSL processors (such as one developed by Lotus Development Corporation (TM)) typically use the DOM during the process of transforming an input XML description to an output form for display using a Web browser. The transformation is defined using a set of XSL rules which consist of two parts: a pattern matching description to identify structures within the input XML and a transformation which maps the input matched pattern to an output representation. The preferred embodiment of the present invention advantageously uses the XSL rules in order to transform documents from a non-XML format into an output form for display using a Web browser. This, in effect, extends the XML-based mechanism into a general purpose transformation engine. For example, this can be used to transform a COBOL input structure to an XML tagged output structure. In this latter example, the XSL rules used by the XSL processor 403 are written so that a COBOL document, which has been converted into an object model by COBOL parser 401B and stored into memory 402, is transformed into an XML tagged output structure by passing the output of the memory 402 through the XSL processor 403.

Accordingly, as standard tools become available for creating and managing XSL rules, these tools may be used without change to transform other non-XML structured documents.

Besides being embodied in a data processing apparatus and a data processing method, examples of which are illustrated in Figs. 2-4, the present invention can also be embodied as a computer program product for use with a computer system. Such an implementation may comprise a series of computer readable instructions either fixed on a tangible medium, such as a computer readable media, e.g., diskette, CD-ROM, ROM, or hard disk, or transmittable to a computer system, via a modem or other interface device, over either a tangible medium, including but not limited to optical or analog communications lines, or intangibly using wireless techniques, including but not limited to microwave, infrared or other transmission techniques. The series of computer readable instructions embodies all or part of the functionality previously described herein.

Those skilled in the art will appreciate that such computer readable instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Further, such instructions may be stored using any memory technology, present or future, including but not limited to, semiconductor, magnetic, or optical, or transmitted using any communications technology, present or future, including but not limited to optical, infrared, or microwave. It is contemplated that such a computer program product may be distributed as a removable media with accompanying printed or electronic documentation, e.g., shrink wrapped software, pre-loaded with a computer system, e.g., on a system ROM or fixed disk, or distributed from a server or electronic bulletin board over a network, e.g., the Internet or World Wide Web.

CLAIMS

1. A data processing apparatus comprising:

a memory unit;

system software means, including a plurality of parsers, for implementing a Document Object Model application programming interface, including means for receiving documents from an originating application and processing means for processing received documents using one of the plurality of parsers in order to convert each received document into an object model; and

system software means for storing each object model into the memory unit;

wherein the processing means sends a received document to a particular one of the plurality of parsers depending on a format of the received document.

2. The apparatus of claim 1 wherein one of the parsers is an XML parser and a received XML format document is sent to the XML parser for processing.

3. The apparatus of claim 1 wherein one of the parsers is a COBOL parser and a received COBOL format document is sent to the COBOL parser for processing.

4. The apparatus of claim 1 wherein each document has data and structure associated therewith and the object model is represented as a tree of nodes containing the data and structure of the corresponding document.

5. The apparatus of claim 1 further comprising an XSL processor.

6. The apparatus of claim 5 wherein the XSL processor includes XSL rules for transforming a document from one format to another format.

7. The apparatus of claim 6 wherein the XSL rules are for transforming a document from COBOL format into XML format.

8. The apparatus of claim 1 wherein at least one of the parsers is a lazy parser.

9. A data processing method comprising steps of:

receiving a document from an originating application via a Document Object Model application programming interface which includes a plurality of parsers;

determining a format of the received document;

selecting one of the plurality of parsers depending on the results of the determining step;

converting the received document into an object model using the parser selected at the selecting step; and

storing the object model into a memory unit.

10. The method of claim 9 wherein one of the parsers is an XML parser and a received XML format document is sent to the XML parser for processing.

11. The method of claim 9 wherein one of the parsers is a COBOL parser and a received COBOL format document is sent to the COBOL parser for processing.

12. The method of claim 9 wherein each document has data and structure associated therewith and the object model is represented as a tree of nodes containing the data and structure of the corresponding document.

13. The method of claim 9 further comprising a step of outputting the stored object model from the memory unit to an XSL processor.

14. The method of claim 13 wherein the XSL processor includes XSL rules for transforming a document from one format to another format.

15. The method of claim 14 wherein the XSL rules are for transforming a document from COBOL format into XML format.

16. The method of claim 9 wherein at least one of the parsers is a lazy parser.

17. A computer program product stored on a computer readable carrier medium for, when run by a computer, carrying out a data processing method comprising steps of:

receiving a document from an originating application via a Document Object Model application programming interface which includes a plurality of parsers;

determining a format of the received document;

selecting one of the plurality of parsers depending on the results of the determining step;

converting the received document into an object model using the parser selected at the selecting step; and

storing the object model into a memory unit.

18. The computer program product of claim 17 wherein one of the parsers is an XML parser and a received XML format document is sent to the XML parser for processing.

19. The computer program product of claim 17 wherein one of the parsers is a COBOL parser and a received COBOL format document is sent to the COBOL parser for processing.

20. The computer program product of claim 17 wherein each document has data and structure associated therewith and the object model is represented as a tree of nodes containing the data and structure of the corresponding document.

21. The computer program product of claim 17 further comprising a step of outputting the stored object model from the memory unit to an XSL processor.

22. The computer program product of claim 21 wherein the XSL processor includes XSL rules for transforming a document from one format to another format.

23. The computer program product of claim 22 wherein the XSL rules are for transforming a document from COBOL format into XML format.

24. The computer program product of claim 17 wherein at least one of the parsers is a lazy parser.

25. A data processing apparatus comprising:

means for receiving a document from an originating application via a Document Object Model application programming interface which includes a plurality of parsers;

means for determining a format of the received document;

means for selecting one of the plurality of parsers depending on the results of the determining step;

means for converting the received document into an object model using the parser selected at the selecting step; and

means for storing the object model into a memory unit.

26. The apparatus of claim 25 wherein the document is received from the originating application over the Internet.

27. The apparatus of claim 1 wherein the document is received from the originating application over the Internet.

28. The method of claim 9 wherein the document is received from the originating application over the Internet.

29. The computer program product of claim 17 wherein the document is received from the originating application over the Internet.

PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS
The Patent Office

INVESTOR IN PEOPLE

11.

Application No: GB 9929936.4
Claims searched: 1-29

Examiner: Geoff Western
Date of search: 22 August 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4A (AKBX AUDB)

Int Cl (Ed.7): G06F 17/21 17/22 17/27 17/30

Other: Online : COMPUTER, EPODOC, INSPEC, Internet, JAPIO, TDB, WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	WO 98/53393 A1 (ADOBE SYSTEMS) See whole document	-
X	"Class com.studiom.dom.DOM", as last modified 26 August 1999, at www.studiom.com/sw/dom/docs/api/com.studiom.dom.DOM.html	1-4, 8-12, 16-20, 24-29

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		

jefflj@jefft21

Job Title: » **Re: APPROVAL REQUESTED: IBM a**

Time Queued: Wed Nov 13 14:31:21 2002
Time Printed: Wed Nov 13 14:31:35 2002

Submitted By: jefflj@jefft21

Need help? Call the HelpDesk at 1-888-IBM-HELP.

Or try the WEB pages at: http://w3-1.ibm.com/tools/print/

jefflj@jefft21

PIO_DEV: ibm@usrtpl3a.raleigh.ibm.com#ibmNetPrinter (ibm1130) @ usrtpprt24

FLAG VALUES:
a=0, d=s, f=, j=1, q=300, u=2, ==0, A=1, B=gn, C=!, E=0, H=, J=+, N=1,
O=3, P=PD_ps_TLasFb:queue_device, Q=1, S=!, W=1, X=, Y=0, Z=!

From: Geraldine Y Kan on 11/13/2002 01:36 PM

To: Jeffrey L Jefferson/Raleigh/IBM@IBMUS
cc: Scott Ferrauiola/White Plains/IBM@IBMUS
From: Geraldine Y Kan/Somers/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: IBM and Sharp Team on Enterprise Class, Linux-based Mobile Solution
Importance: Urgent

Hi Jeff

Scott asked that I route the Sharp/ IBM release to you for approval. We're hoping to start pitching this tomrrow morning. Thanks much.

Draft Release: IBM and Sharp Team on Enterprise Class, Linux-Based Mobile Solution
Draft Release: IBM and Sharp Team on Enterprise Class, Linux-Based Mobile Solution

IBM and Sharp Team on Enterprise Class, Linux-Based Mobile Solution

Openness and Flexibility Key to Creating the An Extensive Embedded Linux® Deployment for the Enterprise Market

Armonk, NY and Osaka, Japan. Nov. 15, 2002 – Building on the growing momentum in workforce mobility and corporate investment in Linux®, IBM and Sharp Corporation today announced that they will bring together IBM software and Sharp technologies in a Linux based mobile solution that makes it easier for businesses to extend enterprise applications to mobile workers.

The new Enterprise Edition Zaurus, Sharp's Linux based PDA that includes a foundation of IBM's infrastructure software, will offer seamless, wireless access to a wide range of enterprise applications on an open, flexible platform. This agreement is also IBM's largest end-to-end embedded Linux deployment.

Targeted at an increasingly mobile workforce in which connectivity and access to corporate information is crucial, the Enterprise Edition Zaurus will allow mobile employees to easily access databases to check inventory and order supplies, as well as manage e-mail, address book entries and calendars while on the road. It will also enable companies to access a wide range of applications from a variety of vendors. Sharp and IBM plan to support speech recognition and multi-modal capabilities in future editions.

The announcement creates new opportunities for companies to deploy applications more rapidly by using the flexible, open-standards based Linux and JavaTM platforms. Companies will also have access to a rapidly expanding community of developers building applications that enable mobile access to enterprise data.

"As we move into the On Demand age, companies need to access information, applications and services on their own terms – when, where, and how they choose," said Rod Adkins, GM, IBM Pervasive Computing Division."Customers also expect new solutions and new devices to integrate seamlessly with their infrastructure. IBM and Sharp are working to leverage the flexibility of Linux® and JavaTM - tying together device technologies and solutions to help enterprises extend their infrastructure to make access simpler and interaction easier."

The collaboration with Sharp is a continuation of IBM's strategy to develop and deliver end-to-end pervasive computing solutions for enterprises and mobile workers across different platforms. The move also extends Sharp's larger corporate strategy to create an open standards Linux /Java platform with back-end support for various enterprise applications and databases including IBM's DB2, as well as those from other vendors such as Microsoft®, Oracle® and Sybase®.

"Sharp is committed to delivering a new generation of mobile computing solutions to the enterprise customer that integrate world class hardware functionality with breakthrough software capability," according to Dr. Sakai, Executive Director and General Manager of Sharp's Communications Systems Group. "Sharp's vision is to extend the reach of the Zaurus Linux and Java-based OS platform to the mobile workers' every day environment, seamlessly delivering critical data regardless of location."

Demonstrating IBM's growing momentum in pervasive computing and continued investment in Linux®, IBM will provide end-to-end Linux® and JavaTM-based device infrastructure middleware that gives employees access to corporate applications, information, intranets and databases while they are in the field or on the move and also provides seamless connectivity. Solutions using the Enterprise Edition Zaurus will include IBM's WebSphere Everyplace Access (WEA) and WebSphere Everyplace Connection Manager (WECM) middleware. In addition, IBM offers a full range of it's open source based WebSphere Studio family of development tools for building applications. By using an open framework, developers have the flexibility to build and differentiate products regardless of platform, which in turn could reduce the time to market.

As one of the pioneers of handheld devices, Sharp has introduced innovative features to the mobile computing category, including an integrated sliding QWERTY keyboard, dual expansion slots (Compact FlashTM and SD/MMC), and an open standards Linux/Java application platform. Sharp has demonstrated the potential of this open standards platform by running Apache web server and mySQL database on the Zaurus, and will continue to optimize the Zaurus for mobile enterprise solutions and wireless connectivity.

This Enterprise Edition Zaurus is expected to be available by mid-2003 in the US, followed by rollouts to additional markets worldwide.

###

About Sharp

Sharp Corporation is a worldwide developer of innovative products and core technologies that play a key role in shaping the future of electronics. As a leader in liquid crystal displays (LCDs) and digital technologies, Sharp offers one of the broadest and most advanced lines of consumer electronics, information products and electronic components, while also creating new network businesses.

Sharp Corporation currently employs about 57,000 people worldwide and recorded consolidated annual sales of over 1,803.7 billion yen for the fiscal year ended March 31, 2002. For more information, please visit Sharp's Web site at http://sharp-world.com/index.html.

About IBM

IBM is the world's leading e-business company offering a wide range of services, solutions and technologies that help businesses take full advantage of emerging innovation. IBM's pervasive computing and mobile Internet strategy is to extend e-business applications to the new class of client devices. This involves building, deploying and developing mobile applications by partnering with key industry players, developing groundbreaking initiatives to set open industry standards; and deploying a Business Innovation Services team with hundreds of wireless consultants. IBM also makes chips for a wide range of devices from the world's most powerful computers to the smallest cell phones. IBM can be found on the Web at www.ibm.com.

Geraldine Kan
Public Relations, IBM Pervasive Computing
Wk: 1-914-766-3284 (IBM tie-line 826 3284)
Cell: 1-917-478-5148
gkan@us.ibm.com
IBM MD 1106, Route 100
Somers, NY 10589

----- Forwarded by Geraldine Y Kan/Somers/IBM on 11/13/2002 01:31 PM -----

Scott Ferrauiola To: Geraldine Y Kan/Somers/IBM@IBMUS

cc: Stephen Miller/Raleigh/IBM@IBMUS, Geraldine Y Kan/Somers/IBM@IBMUS
From: Scott Ferrauiola/White Plains/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342

kathryn, haven't reviewed it quite yet. should get to it soon.

can you send me the sharp agreement that supports the announcement and let me know who in legal has been working on it. i can't approve the press release without knowing we have an agreement that has been approved by legal.

Regards.

Scott Ferrauiola
Counsel for Pervasive Computing Division
Route 100, Somers, NY 10589
Tel(Fax): 914-766-1125 (8160) - - Tieline 826
E-Mail: scottfer@us.ibm.com
CONFIDENTIAL - PREPARED BY IBM ATTORNEY- DO NOT FORWARD WITHOUT PRIOR APPROVAL

Kathryn L Parsons

Kathryn L Parsons
11/13/2002 09:24 AM

To: Scott Ferrauiola/White Plains/IBM@IBMUS
cc: Stephen Miller/Raleigh/IBM@IBMUS
From: Kathryn L Parsons/Fort Lauderdale/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342

Scott, Have you had an opportunity to review this eval license yet? Your approval is required because the license is a 2-way cross license which we do not have pre-approval to use. Please let me know if you have any questions.

Regards,

Kate Parsons
IBM Contract Administrator

International Business Machines
8051 Congress Avenue
Boca Raton, FL 33487
Ph: 561-862-2054, Fax: 561-862-3959
--------------------- Forwarded by Kathryn L Parsons/Fort Lauderdale/IBM on 11/13/2002 09:20 AM ---------------------------

Kathryn L Parsons
10/31/2002 08:59 AM

To: Scott Ferrauiola/White Plains/IBM@IBMUS
cc:
From: Kathryn L Parsons/Fort Lauderdale/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342

Scott, I understand I should have submittted this to you for review/approval (see below).

Regards,

Kate Parsons
IBM Contract Administrator

International Business Machines
8051 Congress Avenue
Boca Raton, FL 33487
Ph: 561-862-2054, Fax: 561-862-3959

--------------------- Forwarded by Kathryn L Parsons/Fort Lauderdale/IBM on 10/31/2002 08:57 AM ---------------------------

Yolanda W Rabun
10/31/2002 08:17 AM

To: Kathryn L Parsons/Fort Lauderdale/IBM@IBMUS
cc:

From: Yolanda W Rabun/Raleigh/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342
Importance: Normal

This belongs to Scott F. He handles all Portal products. Here is a responsible atty list, in case this helps:

Regards, Yolanda
Staff Counsel, IBM Legal Department

PRIVILEGED & CONFIDENTIAL / DO NOT FORWARD W/O CONSENT
Visit the RTP Legal Web Site at http://w3legal.raleigh.ibm.com/index.html for answers to common legal questions.
VOICE: (919) 254-9563 FAX: (919) 254-4330
ALTERNATE VOICE: (919) 845-9500 FAX: (845) 491-3109
IBM Software Group, RTP, NC 27709 - INTERNET: YRABUN@US.IBM.COM

Kathryn L Parsons

Kathryn L Parsons
10/29/2002 09:53 AM

To: Dana O'Neal/Austin/IBM@IBMUS, Yolanda W Rabun/Raleigh/IBM@IBMUS
cc: Stephen Miller/Raleigh/IBM@IBMUS
From: Kathryn L Parsons/Fort Lauderdale/IBM@IBMUS
Subject: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342

Dana and Yolanda, Attached for your review and approval is a two-way, inbound and outbound, evaluation license agreement for Cysive as requested by Steve Miller for the companies to provide the following:

Code to be provided by IBM: WebSphere Portal Server which including IBM's WebSphere Application Server, Version 4.2.
Code to be provided by Cysive: Cysive Cymbio Interaction Server, Version 2.3.

All code is generally available.

Yolanda, as a base I used the 2 way eval license we recently did for Whirlpool.

Please provide your comments and/or approval.

Regards,

Kate Parsons
IBM Contract Administrator

International Business Machines
8051 Congress Avenue
Boca Raton, FL 33487
Ph: 561-862-2054, Fax: 561-862-3959

Cysive 2-way Eval.doc has been removed from this note on October 31 2002 by Yolanda W Rabun

*******Attachment(s) have been removed*******

11/13/2002 12:58 PM

cc:
From: Scott Ferrauiola/White Plains/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342

please have the lawyer who negotiated the deal with sharp, jeff jefferson, review and approve the press release. thanks.

Regards.

Scott Ferrauiola
Counsel for Pervasive Computing Division
Route 100, Somers, NY 10589
Tel(Fax): 914-766-1125 (8160) - - Tieline 826
E-Mail: scottfer@us.ibm.com
CONFIDENTIAL - PREPARED BY IBM ATTORNEY- DO NOT FORWARD WITHOUT PRIOR APPROVAL

----- Forwarded by Scott Ferrauiola/White Plains/IBM on 11/13/2002 12:57 PM -----

Kathryn L Parsons

11/13/2002 12:02 PM

To: Scott Ferrauiola/White Plains/IBM@IBMUS
cc: Geraldine Y Kan/Somers/IBM@IBMUS
From: Kathryn L Parsons/Fort Lauderdale/IBM@IBMUS
Subject: Re: APPROVAL REQUESTED: Two-Way Evaluation License for Cysive No. 4902FA1342

Scott, I was trying to determine which Sharp SOW your press release falls under - I'm having to guess (in light of no response), but I believe it's under the one attached below (base agreement also attached). Both have been fully executed and the attorney who approved was Jeff Jefferson. The agreement numbers are:

Base Agreement No.:	4902FA1147
SOW No.:	4902FA1148

Clean Copy Final Sharp-IBM_MCA_Base_Rev9_B10: Clean Copy Final Sharp-SOW9_B103.c

Regards,

Kate Parsons
IBM Contract Administrator

International Business Machines
8051 Congress Avenue
Boca Raton, FL 33487
Ph: 561-862-2054, Fax: 561-862-3959

SCOTT FERRAUIOLA
11/13/2002 10:55 AM

To: Kathryn L Parsons/Fort Lauderdale/IBM@IBMUS